MA 3232 - Numerical Analysis
Objectives and Course Overview

I. Introduction

This course responds to the needs of the engineering and physical sciences curricula by providing an applications-oriented introduction to numerical methods/analysis. Rather than a pure discussion and analysis of methods, we shall often integrate a discussion of the properties of engineering and physical problems with the discussion of methods by which such problems may be solved numerically. This approach is more "natural" and more like the one students actually follow when applying numerical methods within their areas of interest.

Topics in function approximation, nonlinear equations, interpolation, numerical integration and differentiation, and numerical solution of ordinary differential equations will be similarly treated. The discussion of approximate arithmetic and error propagation will also arise in a natural way.

II. Objectives

Upon successful completion of this course, one should be able to:

1. Describe the general difficulties that arise because most engineering and scientific analysis uses only approximate models and data, and because most computers usually use finite precision, often non-decimal arithmetic.

2. State the definition of a normalized floating-point number. Given an appropriate real number, convert it to normalized, decimal floating-point form using a specified number of digits and either chopping or rounding arithmetic, and identify the most and least significant digits in that number.

3. State the definition of machine precision, and briefly describe its importance with respect to engineering and scientific calculation. Given the base and number of digits in a floating-point number system, calculate machine precision in that system for either chopping or rounding arithmetic.

4. List the primary sources of error in floating-point computation. Properly simulate appropriate decimal, floating-point calcluations using either chopping or rounding. Be able to identify when *catastrophic cancellation* may occur in a given computation, and, if possible, how to avoid it.

5. State the definitions of forward error and backward error in numerical computation. Given a specified problem, determine the relevant forward and backward errors.

6. State the definition of the condition number of a problem, and describe, in general terms, how it relates to how the sensitivity of solutions to that problem to inaccuracies in the original problem data. Given an appropriate problem, determine its condition number, and, based on that number, discuss, quantitatively, the error behavior we should expect in numerical solution of that problem.

7. State what is meant by backward stability of an algorithm. Simulate an given appropriate algorithm, using either rounded or chopped decimal, floating-point arithmetic, and, based on your calculated result, describe whether or not the algorithm appears to be backward stable.

8. State the fundamental equation that relates the expected accuracy of the solution to a given problem to the condition of that problem, the backward stability of the algorithm we are using to solve that problem, and the expected inaccuracy in the original problem data.

9. Apply the methods of bisection, false position, secant, Newton and linear iteration (i.e., $x_{n+1} = g(x_n)$) to solve a given nonlinear equation. State general conditions which guarantee convergence of each of these methods, and determine whether or not those conditions are satisfied in given cases.

10. State the definition of the order of convergence of an iterative method, and describe how the order of a method relates to both the convergence and error behavior for that method. Describe the order of convergence of standard iterative methods.

11. State the basis for Aitken acceleration, and apply the method in a given case.

12. Apply the Newton-Raphson method to solve systems of nonlinear equations.

13. State the definitions of interpolation. Describe why most interpolation is based on polynomials, but why Taylor polynomials are not generally used.

14. Given an appropriate set of data, construct an Lagrange interpolation polynomial of a specified degree. State the general advantages and disadvantages of Langrange polynomials.

15. State the definition of the forward difference operator. Using that definition and a given set of data, construct, either numerically or symbolically, an appropriate difference table and Newton-Gregory forward difference interpolation polynomial for a given set of data.

16. State and use the formulas for bounding the error in polynomial interpolation based on derivatives, and for estimating error based on differences.

17. Describe the basic ideas and principles behind cubic spline interpolation, and its limitations.

18. State the basic ideas and principles behind least squares function approximation. Given an appropriate function, compute a specificed, low-degree Taylor polynomial least squares fit to that function. State the primary drawbacks to using high-degree Taylor polynomials for least squares. Briefly describe the role of orthogonal polynomials in least squares.

19. Describe the basic ideas and principles behind rational approximation, and its limitations.

20. Construct and apply formulas for approximating specified derivatives of functions by differentiation of appropriate interpolation polynomials.

21. Construct local and composite (global) quadrature formulas based on integration of appropriate interpolation polynomials with evenly-spaced nodes.

22. Apply trapezoidal and Simpson's Rule quadratures in composite form to find the approximate value of an integral.

23. State the idea behind the derivation of Gauss quadrature rules, and why they often work very well. Apply given low order Gauss quadrature.

24. Describe the basic concepts behind Runge-Kutta (R-K) methods for the initial value problem (IVP) and apply specific R-K methods in given problems.

25. State how multistep methods are derived and apply multistep predictor-corrector methods in given problems.

26. State the advantages/disadvantages of single step and multistep methods for the IVP and when use of each is recommended.

27. Describe what the terms convergence, stability and consistency mean in the context of numerical solution ordinary differential equations. Explain the possible instability of multistep methods.

28. State the definition of the *order* of an interpolation, differentiation, or integration method. Describe the relationships between the order of a given method and error estimation. Use the knowledge of the order of given methods to estimate error by either extrapolation or the "next term" paradigm.

29. Describe the basic concepts behind and differences between shooting, finite difference, Galerkin, finite-element, and Rayleigh-Ritz methods for ordinary differential equation boundary value problems, and apply each in appropriate problems.

30. Given library routines for the solution of the above problems as well as other utility routines, be able to indicate how to use them in the solution of a larger problem involving the solution of several 'subproblems'.

    For example:

    Assume that routines for the solution of (1) a nonlinear equation and (2) an ordinary differential equation on the interval $[a, b]$, as well as hardware and software for producing (3) appropriate graphical output (e.g. curve plots). (Specific information about input, output, and other requirements for these routines will be given.) The student may then be asked to outline (in English, MATLAB, flow charts, mathematical symbols, or a combination thereof) a program for solution of the following problem:

    With $y(x)$ defined by

$$y' = f(x, y)$$
$$y(a) = y_0 \ ,$$

    find the smallest $x* > a$ such that $y(x*) = 0$, and plot y versus x on $[a, x*]$.

III. Assignments, Examinations and Grading

There will be generally be at least one in-class hour exam (midterm) plus a comprehensive two-hour final. These exams will, in total, comprise up to 50% of the course grade, and will be individually weighted according to their length.

In addition, there will graded computer projects, which are designed to both illustrate and emphasize certain aspects of software and algorithms, and to expose the student to reasonably realistic engineering and scientific computation problems using state-of-the-art software. The combined grades on these programming assignments will determine the remainder of the course grade.

Lastsly, there will also be a number of numerical laboratories conducted as instructor-led demonstrations. These laboratories will not have any associted required gradable assignments that must be completed. However, understanding of much of the material in these laboratories is relevant to both the examinations and the graded programming assignments. Therefore, and students are strongly encouraged to both complete the associated laboratory worksheets during the in-class demonstration, and to rerun outisde of class any portions where they are unsure of the proper interpretation of the results presented in class.

IV. Conclusion

The analysis of numerical methods is important because virtually all of the students in this course will use the computer methods to solve engineering and scientific problems during their studies; for many of them computational results will constitute a major portion of their thesis. This course not only examines methods, but also provides an opportunity to gain computational experience in the use of high-quality library software through the programming assignments made during the course. These assignments are designed to illustrate the theory and provide experience in obtaining numerical solutions of engineering and scientific problems. In addition, some of the assignments are used to exhibit common computational difficulties and ways of avoiding them.

12/04
Zh